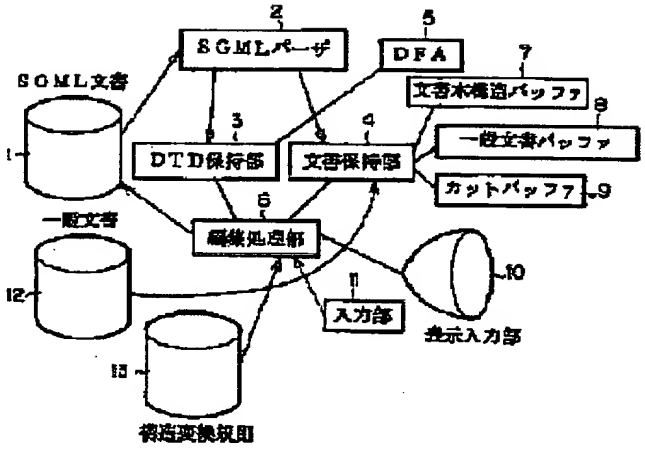
STRUCTURED DOCUMENT EDITING DEVICE

Patent number:	JP8016576
Publication date:	1996-01-19
Inventor:	SENDA SHIGEYA
Applicant:	RICOH KK
Classification:	
- international:	***G06F17/24;*** **G06F17/24**; (IPC1-7): G06F17/24
- european:	
Application number:	JP19940150868 19940701
Priority number(s):	JP19940150868 19940701

Abstract of **JP8016576**

PURPOSE:To efficiently perform an editing work by eliminating a substitution work by confirming the substitution of a real reference by a user afterward by substituting the real reference when a paste is performed. CONSTITUTION:An editor receives an SGML document 1 as an input and stores an edition result as an output by an SGML document form. In the case of a new document, a DTD is designated by a user and the only DTD is read. When the editor reads the SGML document/DTD, the input is analyzed by an SGML parser 2 at first. The SGML parser 2 prepares a finite automata (DFA) 5 corresponding to the DTD as an analysis result and stores the automata in a DTD holding part 3. The DFA 5 expresses a list where the lower rank node of each tag is possible. When the character string matching with the real declaration in an insertion destination document exists in a text area when the text area is inserted into a structured document, the character string is substituted for a real reference.



a)

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平8－16576

(43)公開日　平成８年(1996)１月19日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　17/24				
		9288－5L	Ｇ０６Ｆ　15/ 20	５５４　Ｈ

審査請求　未請求　請求項の数６　ＯＬ　（全 ９ 頁）

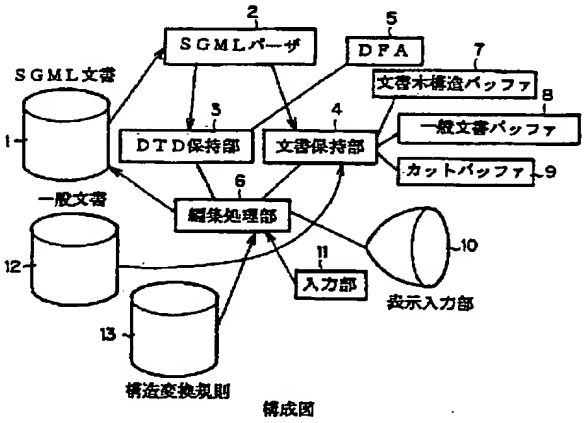
(21)出願番号　特願平6－150868

(22)出願日　平成６年(1994)７月１日

(71)出願人　000006747
株式会社リコー
東京都大田区中馬込１丁目３番６号
(72)発明者　千田　滋也
東京都大田区中馬込１丁目３番６号　株式会社リコー内
(74)代理人　弁理士　高野　明近

(54)【発明の名称】　構造化文書編集装置

(57)【要約】
【目的】　ペーストする際に実体参照に置き換えることにより、ユーザがあとから確認して置き換える作業がいらなくなるようにし、編集作業の効率化を図る。
【構成】　エディタはＳＧＭＬ文書１を入力として受取り、編集結果を出力としてＳＧＭＬ文書形式で格納する。新規文書の場合、使用者にＤＴＤを指定してもらい、ＤＴＤのみを読み込む。エディタがＳＧＭＬ文書・ＤＴＤを読み込む場合、まず入力をＳＧＭＬパーザ２で解析する。該ＳＧＭＬパーザ２は解析結果としてＤＴＤに対応して有限オートマトン（ＤＦＡ）５を作成し、ＤＴＤ保持部３に格納する。ＤＦＡ５は各タグの下位のノードの可能な並びを表現する。構造化文書中にテキスト領域を挿入する際に、挿入先文書にある実体宣言に一致する文字列がテキスト領域にある場合、該文字列を実体参照に置き換える。



構成図

1

【特許請求の範囲】

【請求項１】　構造化文書を保持する文書保持部と、該構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、構造化文書中にテキスト領域を挿入する際に、挿入先文書にある実体宣言に一致する文字列がテキスト領域にある場合、該文字列を実体参照に置き換えることを特徴とする構造化文書編集装置。

【請求項２】　構造化文書を保持する文書保持部と、該構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、構造化文書中にテキスト領域を挿入する際に、挿入元文書にある実体宣言が未定義である場合、該実体宣言を挿入先文書に複写することを特徴とする構造化文書編集装置。

【請求項３】　構造化文書を保持する文書保持部と、構造に関する情報を保持するＤＴＤ保持部と、編集過程の情報を表示する表示入力部と、構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、構造化文書中の部分構造を対象として削除、複写、挿入を行うことを特徴とする構造化文書編集装置。

【請求項４】　前記部分構造を挿入する際、挿入先文書の文書構造に適合するノード位置を表示することを特徴とする請求項３記載の構造化文書編集装置。

【請求項５】　前記部分構造を挿入した際、挿入した部分構造中の文書構造に適合しないノードの情報を表示することを特徴とする請求項３記載の構造化文書編集装置。

【請求項６】　前記部分構造を挿入する際、挿入元と挿入先の構造間の変換規則により変換を行った後に挿入することを特徴とする請求項３記載の構造化文書編集装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、構造化文書編集装置に関し、より詳細には、ペースト（paste）する際に実体参照に置き換えることにより、ユーザがあとから確認して置き換える作業がいらなくなるようにし、編集作業の効率化を図るようにした構造化文書編集装置に関する。

【０００２】

【従来の技術】従来の構造文書編集装置について記載した公知文献としては、例えば、特開平２－２４７７７２号公報がある。この公報のものは、ユーザがタブ表示を必要とするかどうかに係わりなく、文書階層の要素論理構造を操作することを可能にし、標準一般化マークアップ言語（ＳＧＭＬ）によって定義されるようなテキストやグラフィック・イメージ又は他のデータを含む階層的文書構造の要素をマークするものである。すなわち、構造文書編集において、カーソルのある位置のエレメントを含む構造の階層を表示するものである。

2

【０００３】ＳＧＭＬ規格は、構造化文書の交換を標準化するための規格である。文書の構造は文書型定義（ＤＴＤ，Document Type Definition）によって定義される。ＤＴＤで規定された文書構造にしたがって文書を作成する場合、プレーンテキストとしてタグを内容と同時に入力し、バッチ処理によってＳＧＭＬ文書を処理するか、専用のエディット機能を持つ構造化文書エディタを利用する必要がある。

【０００４】

【発明が解決しようとする課題】文書を標準化し、多量の文書を効率良く管理、運用するために構造化文書（ＳＧＭＬ）を用いることが考えられる。しかし、現状ではＳＧＭＬ文書ではなく他の記述形式で書かれた文書、例えば、通常のワードプロセッサなどで書かれた文書などがほとんどで、それらをＳＧＭＬ化する作業が必要である。これらの作業のために文書の変換処理を行う機構が考えられる。変換機構は、文書構造がはっきり分かっている場合はバッチ的に処理可能である場合もあるが、そうでない場合、人間の介入が必要になって来る。従来のエディタでは、従来文書からの変換に関してはほとんど考慮されておらず、単にテキストのcut&pasteなどが用意されているだけであった。

【０００５】本発明は、このような実情に鑑みてなされたもので、ペーストする際に実体参照に置き換えることにより、ユーザがあとから確認して置き換える作業がいらなくなるようにし、編集作業の効率化を図ること、また、ペーストする際に未定義の実体宣言を複写先にコピーすることにより、ユーザがあとから新たに宣言する必要をなくし、編集作業の効率化を図ること、さらに、部分構造を構造文書間で削除、複写、挿入することで、同様の内容を新たに作ることなく文書を作成でき、編集作業の効率化を図るようにした構造化文書編集装置を提供することを目的としている。

【０００６】

【課題を解決するための手段】本発明は、上記課題を解決するために、（１）構造化文書を保持する文書保持部と、該構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、構造化文書中にテキスト領域を挿入する際に、挿入先文書にある実体宣言に一致する文字列がテキスト領域にある場合、該文字列を実体参照に置き換えること、或いは、（２）構造化文書を保持する文書保持部と、該構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、構造化文書中にテキスト領域を挿入する際に、挿入元文書にある実体宣言が未定義である場合、該実体宣言を挿入先文書に複写すること、或いは、（３）構造化文書を保持する文書保持部と、構造に関する情報を保持するＤＴＤ保持部と、編集過程の情報を表示する表示入力部と、構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備

し、構造化文書中の部分構造を対象として削除、複写、挿入を行うこと、更には、（４）前記（３）において、前記部分構造を挿入する際、挿入先文書の文書構造に適合するノード位置を表示すること、更には、（５）前記（３）において、前記部分構造を挿入した際、挿入した部分構造中の文書構造に適合しないノードの情報を表示すること、更には、（６）前記（３）において、前記部分構造を挿入する際、挿入元と挿入先の構造間の変換規則により変換を行った後に挿入すること、或いは、（７）一般文書と構造化文書を保持する文書保持部と、構造に関する情報を保持するＤＴＤ保持部と、編集過程の情報を表示する表示入力部と、文書の編集処理を行う編集処理部とを具備し、一般文書の内容を削除、複写し、その内容を挿入する際に構造文書の注目点の次に来るデータ内容までの要素列を内容モデルから作成し、該内容モデルを使用者にリストとして示して選択させ、その要素列を作成して一般文書の内容をデータ内容として挿入することを特徴としたものである。

【０００７】

【作用】前記構成を有する本発明の構造化文書編集装置は、構造化文書を保持する文書保持部と、該構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、（１）構造化文書中にテキスト領域を挿入する際に、挿入先文書にある実体宣言に一致する文字列がテキスト領域にある場合、該文字列を実体参照に置き換えるので、ペーストする際に実体参照に置き換えることにより、ユーザがあとから確認して置き換える作業がいらなくなるため、編集作業が効率的に行える。（２）構造化文書中にテキスト領域を挿入する際に、挿入元文書にある実体宣言が未定義である場合、該実体宣言を挿入先文書に複写するので、ペーストする際に未定義の実体宣言を複写先にコピーすることにより、ユーザがあとから新たに宣言する必要がなく、編集作業が効率的に行える。

【０００８】また、構造化文書を保持する文書保持部と、構造に関する情報を保持するＤＴＤ保持部と、編集過程の情報を表示する表示入力部と、構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、（３）構造化文書中の部分構造を対象として削除、複写、挿入を行うので、部分構造を構造文書間で削除、複写、挿入することで同様の内容を新たに作ることなく文書を作成でき、編集作業が効率的に行える。（４）前記部分構造を挿入する際、挿入先文書の文書構造に適合するノード位置を表示するので、部分構造を構造文書間で削除、複写、挿入する際に内容モデルを壊すことなく挿入できる位置を見付けることができ、編集作業が効率的に行える。（５）前記部分構造を挿入した際、挿入した部分構造中の文書構造に適合しないノードの情報を表示するので、部分構造を構造文書間で削除、複写、挿入した際に内容モデルに違反する内容を見

付けることができ、編集作業が効率的に行える。（６）前記部分構造を挿入する際、挿入元と挿入先の構造間の変換規則により変換を行った後に挿入するので、部分構造を構造文書間で削除、複写、挿入する際に、部分構造を複写先に適合するよう変換することができ、編集作業が効率的に行える。

【０００９】さらに、一般文書と構造化文書を保持する文書保持部と、構造に関する情報を保持するＤＴＤ保持部と、編集過程の情報を表示する表示入力部と、文書の編集処理を行う編集処理部とを具備し、（７）一般文書の内容を削除、複写し、その内容を挿入する際に構造文書の注目点の次に来るデータ内容までの要素列を内容モデルから作成し、該内容モデルを使用者にリストとして示して選択させ、その要素列を作成して一般文書の内容をデータ内容として挿入するので、一般の文書から構造文書への変換の際、構造を簡単に作成することができ、編集作業が効率的に行える。

【００１０】

【実施例】実施例について、図面を参照して以下に説明する。図１は、本発明による構造化文書編集方式の一実施例を説明するための構成図で、図中、１はＳＧＭＬ文書、２はＳＧＭＬパーザ、３はＤＴＤ保持部、４は文書保持部、５はＤＦＡ（有限オートマトン）、６は編集処理部、７は文書木構造バッファ、８は一般文書バッファ、９はカットバッファ、１０は表示入力部、１１は入力部、１２は一般文書、１３は構造変換規則である。

【００１１】エディタの入力はＳＧＭＬ文書であり、ＳＧＭＬ文書１内にはＤＴＤの指定またはＤＴＤそのものが記述されている。エディタはＳＧＭＬ文書を入力として受取り、編集結果を出力としてＳＧＭＬ文書形式で格納する。新規文書の場合、使用者にＤＴＤを指定してもらい、ＤＴＤのみを読み込む。エディタがＳＧＭＬ文書・ＤＴＤを読み込む場合、まず入力をＳＧＭＬパーザ２で解析する。該ＳＧＭＬパーザ２は解析結果としてＤＴＤに対応して有限オートマトン（ＤＦＡ）５を作成し、ＤＴＤ保持部３に格納する。ＤＦＡ５は各タグの下位のノードの可能な並びを表現する。

【００１２】図２は、１つのタグ“kaigi”についてのＤＦＡを示す図である。ＤＴＤで定義される各タグ（エレメント）毎にＤＦＡ５が作成され、ＤＴＤ保持部３にその他のＤＴＤの情報（実体宣言、ＤＴＤ名 etc…）とともに格納される。後の処理では、それを利用してＳＧＭＬ文書１の解析を続ける。ＳＧＭＬ文書自体の解析の結果として内部に、図３に示すような文書構造に対応する木構造を作成する。作成した木構造は元になったＤＴＤと関連づけて文書保持部４に格納される。木構造の各ノードには種別毎に情報が保持されている。

タグ：タグ名、属性

データ：文字列データ

processing instruction：文字列データ

5

【００１３】図４は、図２、図３に対応するＳＧＭＬ文書のＤＴＤの一部を示す図である。また、図５は、図４のＤＴＤを使用した文書の例の一部を示す図である。また、図６は、表示部における画面例を示す図で、この画面では一般文書のウインドウと構造化文書のウインドウが示されている。図１において、カットバッファはcutまたはcopyの操作によって一般文書、文書木構造中のテキストまたは文書木構造の部分構造を保持する。入力部（キーボード、マウス等）によって図６の画面上でテキスト、構造を指定し、コマンドを実行することでカットバッファにその内容を入力する。

【００１４】図７は、カットバッファに内容をpaste（ペースト）する際の動作を説明するためのフローチャートである。以下、各ステップ（step）に従って順に説明する。まず、テキストのみかどうかを判断し（step１）、テキストのみであれば、次に文字列挿入位置を得（step２）、テキスト置換処理を行う（step３）。次に、挿入位置に文字列を挿入する（step４）。前記step１において、テキストのみでなければ、構造挿入位置を得（step５）、ノードのテキスト毎にテキスト置換処理を行う（step６）。次に、挿入位置に構造を挿入する（step７）。

【００１５】この動作は入力部からの指示によって起動される。カットバッファに取り込まれた内容は取り込まれた元の文書の情報と内容を保持している。内容はテキストのみか構造部分木（請求項３）である。そのどちらかによって処理が分かれる。テキストのみの場合、挿入位置を得てテキスト置換処理を行い、その結果を挿入する。部分構造の場合、挿入位置を得て部分構造のテキストを含むノードに対してテキスト置換処理を行う。従来はテキストのみのcut&pasteが行われているのみであったが、請求項３においては、単に文字列のcut&pasteだけでなく、構造単位でcut&pasteでき、便利である。

【００１６】図８は、テキスト置換処理を説明するためのフローチャートである。以下、各ステップ（step）に従って順に説明する。まず、テキスト中にある実体参照があるとき、それが複写先にない場合、複写元の実体宣言を複写先にコピーする（step１１）。次に、複写先で定義されている実体宣言と一致する文字列をその実体参照に置き換える（step１２）。すなわち、元テキスト中に実体参照がある場合で、その実体参照が複写先文書に存在しない場合、それを複写先に登録する（請求項２）。このように、請求項２においては、cutされたテキスト中に実体参照がある場合、元の文書では定義されているが、挿入された文書では未定義である可能性がある。こういった場合に、自動的に挿入先文書に実体定義を複写する機能が考えられる。

【００１７】逆に挿入先文書にある実体宣言と一致する文字列がテキスト中にある場合、それを実体参照に置換する（請求項１）。このように、テキストのcut&paste

6

において、従来は単純に文字列をcut&pasteするだけであったが、その文字列中に複写先において実体宣言がなされている文字列が含まれている場合が考えられる。請求項１においては、これらの文字列を実体参照にpasteの際、置き換えることで挿入先での文書の利用がやりやすくなる。

【００１８】図９は、図７における構造挿入位置を得るための処理を説明するためのフローチャートである。以下、各ステップ（step）に従って順に説明する。まず、一般構造変換処理を行い（step２１）、挿入する構造木のトップノードをｘとする（step２２）。次に挿入先文書のノードを１つとり、ｙとし（step２３）、ｙの子どもまたは兄弟としてｘが来ることができるかＤＦＡをサーチする（step２４）。次に、ｘが来ることができるかどうかを判断し（step２５）、来ることができなければ、前記step２３へ行き、来ることができれば、ノードを兄弟の場合は青、子どもの場合は赤でハイライト表示する（step２６）。

【００１９】次に、全てのノードを調べたかどうかを判断し（step２７）、調べてなければ、前記step２３へ行き、調べてあれば、ユーザに挿入位置を選択してもらう（step２８）。次に、選択位置がＤＴＤに適合するかＤＦＡを検索し（step２９）、来ることができるかどうかを判断し（step３０）、来ることができなければ、警告メッセージを表示する（step３１）。次に、挿入された並びのコンテントモデルをＤＦＡでチェックする（step３２）。挿入された構造のコンテントモデルをＤＦＡで再帰的にチェックし（step３３）、両方のチェックでＤＦＡが満足されなかった位置のノードを反転表示する（step３４）。

【００２０】テキストの場合は、単純にテキスト表示の位置を得るだけであるが、構造の場合、挿入する構造がＤＴＤに適合することが重要である。まず、挿入できる位置をユーザに示すため、木構造をサーチし、挿入可能位置をＤＴＤのＤＦＡから求める。その位置をハイライト表示する（請求項４）。ノードに対し、挿入可能位置は兄弟、または子どもの位置の２通りが考えられる。

【００２１】そこで、この実施例では兄弟の位置の場合ノードに対して青で、子ども位置の場合赤でハイライト表示する。ユーザは挿入先を選択するが、画面上では挿入位置のノードをクリックすることで行う。このときダイアログを表示し、兄弟として挿入するか、子どもとして挿入するかを選択する。このように、請求項４においては、構造のcut&pasteでは構造を挿入先に挿入するとき、ＤＴＤのコンテントモデルに適合するノード位置を表示することで間違った挿入を防ぐことができる。

【００２２】選択されたノードがＤＦＡに適合しない場合、警告を表示する（請求項５）。また、適合する場合でも、挿入によってコンテントモデルが崩れる場合がある。例えば、挿入位置に同じエレメントによる構造が存

7

在する場合である。この場合、同じエレメント名の構造が挿入によって並んでしまう。さらに、挿入された構造が挿入前の文書で正しいコンテントモデルであっても挿入後、挿入先のＤＴＤと適合するとは限らない。そこで、ならびと、挿入された構造をＤＦＡでチェックし、適合しない場合、ノードを反転表示し、使用者に注意を促す。

【００２３】このようなＳＧＭＬ文書間の部分構造のcut&pasteでは、ＤＴＤが異なっている場合、挿入された側の文書に同じelementがない場合や名前が同じでもコンテントモデルが異なる場合が考えられる。請求項５においては、コンテントモデルが適合しない場合、適合しない箇所を表示し、ユーザに変更を促すので便利である。

【００２４】ＤＴＤの構造が元のＤＴＤと挿入先で全く異なる場合、挿入した後、エレメント名の変更や構造の変更を多量に行わなければならない。全く異なる文書でも何らかの意図があってcut&pasteしているわけであるので、使用者にとっては規則的な対応関係があるはずである。そこで、決まっている対応関係がある場合、それを規則的に変更してやれば良い。異なるＤＴＤ文書の変換はＤＳＳＳＬで一般文書変換（ＧＴＬＰ，ＳＴＴＰ）で行われている。これは文書全体を対象とした物であるが、ここではcut&pasteの部分構造木を対象とする。この処理が図９の最初の所で行われる（請求項６）。このように、請求項６においては、頻繁にcut&pasteする場合、自動的に変換できるようにルールを設定できればユーザの手間を省くことができる。

【００２５】一般文書から構造化文書へ変換する場合、cut&pasteだけでは元の文書にない構造を先に変換先に作成しておいてからテキスト内容を挿入する必要があり、変換作業に手間がかかる。そこで、一般文書から構造化文書への変換の際にエディタにおいて変換支援モードを設ける。このモードにおいては、構造化文書中に挿入ポイントというものを設ける。挿入ポイントは構造木のノードに存在する。

【００２６】図１０は、変換支援モードペースト処理を説明するためのフローチャートである。以下、各ステップ（step）に従って順に説明する。まず、テキスト置換処理を行い（step４１）、ノードパスリストｘを空にする（step４２）。次に、挿入ポイントのノードをｙとし（step４３）、ｙの子ども方向にパスを作成してｘに登録する（step４４）。ｙの兄弟方向にパスを作成してｘに登録し（step４５）、ｙの親をｙとする（step４６）。次に、ｙが存在するかどうかを判断し（step４７）、ｙが存在しなければ、前記step４５に戻り、ｙが存在すれば、パスリストを表示してパスを選択させる（step４８）。パスからノードを作成し、カットバッファからテキストをデータノードに挿入する（step４９）。

8

【００２７】すなわち、構造支援モードでは図７で示したpaste処理と動作が異なる。図１０ではまず図７と同様のテキスト置換処理（図８）が行われる。次に、挿入ポイントにおいて次のデータノードにたどり着くノードパス候補を求める。これは、ＤＦＡを挿入ポイントとその祖先のノードからたどることによって得られる。

【００２８】例えば、図３においてkenmeiの所に挿入ポイントがあるとき、パスは

kenmei->sakusei:name：

となる。ただし、“->”は弟を示し、“:”は子どもを示す。つまり、図６にある一般文書で“千田滋也”をcutし、挿入ポイントkenmeiで挿入すると、自動的にパスに従ってsakuseiとnameが作られ、図５に示すようにmynameの実体宣言があれば、実体参照（&myname；）に変換されてデータがnameの下に挿入される。

【００２９】テキスト部分のcut&pasteだけでは他の文書形式から変換する際に不便である。入力文字列に対してユーザがＳＧＭＬの構造として指定する機能を付け加えることにより、より効率的に取り込むことが可能になる。つまり、取り込もうとしている部分に対する構造をユーザが指定する手段、および、システム側から指定を手助けするための候補を与えてやることにより、効率良く構造のない文書を構造を持つ文書構造に変換することができる。

【００３０】

【発明の効果】以上の説明から明らかなように、本発明によると、以下のような効果がある。

（１）請求項１に対応する効果：構造化文書を保持する文書保持部と、該構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、構造化文書中にテキスト領域を挿入する際に、挿入先文書にある実体宣言に一致する文字列がテキスト領域にある場合、該文字列を実体参照に置き換えるので、ペーストする際に実体参照に置き換えることによりユーザがあとから確認して置き換える作業がいらなくなるため、編集作業が効率的に行える。

（２）請求項２に対応する効果：構造化文書中にテキスト領域を挿入する際に、挿入元文書にある実体宣言が未定義である場合、該実体宣言を挿入先文書に複写するので、ペーストする際に未定義の実体宣言を複写先にコピーすることによりユーザがあとから新たに宣言する必要がなく、編集作業が効率的に行える。

（３）請求項３に対応する効果：構造化文書を保持する文書保持部と、構造に関する情報を保持するＤＴＤ保持部と、編集過程の情報を表示する表示入力部と、構造化文書の編集処理を行う編集処理部と、文書内容の一部を蓄えるバッファとを具備し、構造化文書中の部分構造を対象として削除、複写、挿入を行うので、部分構造を構造文書間で削除、複写、挿入することで同様の内容を新たに作ることなく文書を作成でき、編集作業が効率的に

9

行える。

（４）請求項４に対応する効果：前記部分構造を挿入する際、挿入先文書の文書構造に適合するノード位置を表示するので、部分構造を構造文書間で削除、複写、挿入する際に内容モデルを壊すことなく挿入できる位置を見付けることができ、編集作業が効率的に行える。

（５）請求項５に対応する効果：前記部分構造を挿入した際、挿入した部分構造中の文書構造に適合しないノードの情報を表示するので、部分構造を構造文書間で削除、複写、挿入した際に内容モデルに違反する内容を見付けることができ、編集作業が効率的に行える。

（６）請求項６に対応する効果：前記部分構造を挿入する際、挿入元と挿入先の構造間の変換規則により変換を行った後に挿入するので、部分構造を構造文書間で削除、複写、挿入する際に、部分構造を複写先に適合するよう変換することができ、編集作業が効率的に行える。

（７）解決手段７に対応する効果：一般文書と構造化文書を保持する文書保持部と、構造に関する情報を保持するＤＴＤ保持部と、編集過程の情報を表示する表示入力部と、文書の編集処理を行う編集処理部とを具備し、一般文書の内容を削除、複写し、その内容を挿入する際に構造文書の注目点の次に来るデータ内容までの要素列を内容モデルから作成し、該内容モデルを使用者にリストとして示して選択させ、その要素列を作成して一般文書の内容をデータ内容として挿入するので、一般の文書から構造文書への変換の際、構造を簡単に作成することができ、編集作業が効率的に行える。

10

【図面の簡単な説明】

【図１】　本発明による構造化文書編集装置の一実施例を説明するための構成図である。

【図２】　本発明におけるＤＦＡを示す図である。

【図３】　本発明における文書保持部の木構造を示す図である。

【図４】　本発明におけるＤＴＤの例を示す図である。

【図５】　本発明におけるＳＧＭＬ文書例を示す図である。

【図６】　本発明における表示部の画面例を示す図である。

【図７】　本発明におけるペースト処理を説明するためのフローチャートである。

【図８】　本発明におけるテキスト置換処理を説明するためのフローチャートである。

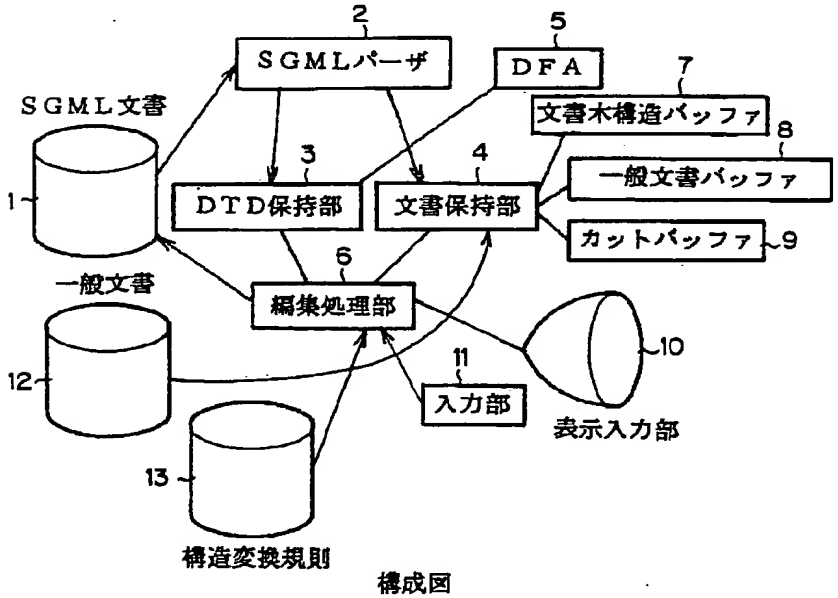
【図９】　本発明における構造挿入位置を保つ処理を説明するためのフローチャートである。

【図１０】　本発明における変換支援モードペースト処理を説明するためのフローチャートである。

【符号の説明】

１…ＳＧＭＬ文書、２…ＳＧＭＬパーザ、３…ＤＴＤ保持部、４…文書保持部、５…ＤＦＡ（有限オートマトン）、６…編集処理部、７…文書木構造バッファ、８…一般文書バッファ、９…カットバッファ、１０…表示入力部、１１…入力部、１２…一般文書、１３…構造変換規則。

【図１】



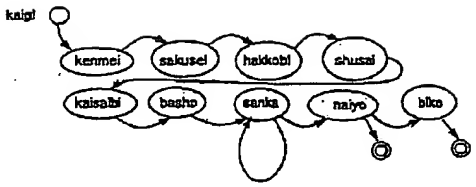
構成図

【図５】

```
<!DOCTYPE kaigi SYSTEM"(DTD)-kaigi"
[
<!ENTITY  ename    "千田稔也">
]>
<kaigi>
<kenmei>MD Cページョンアップについて</kenmei>
<sankasi>
<name>&ename;</name>
<lab>421</lab>
<phone type="short">3714</phone>
<email>senda@ipo.rdc.ricoh.co.jp</email>

</sankasi>
<hakkoh1>
<shusai>
<name>松田 透</name>
<lab>421</lab>
<phone type="short">3710</phone>

        ..................

</kaigi>
```

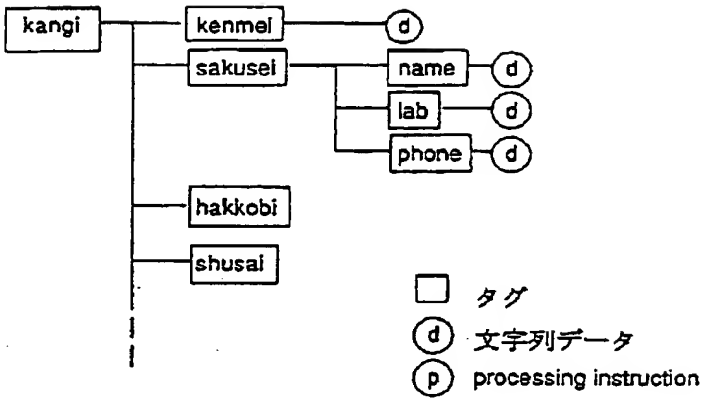
ＳＧＭＬ文書例

【図２】



DFA

【図３】



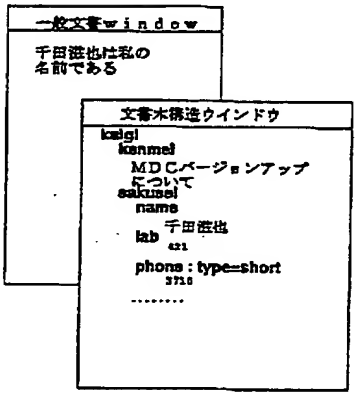
文書保持部の木構造

【図４】

```
<!-This is a document type definition for "会議開催通知" ->

<!ENTITY  % person SYSTEM "(DTD)-prs" -include 人名所属 entity ->
<!ENTITY  % bun    SYSTEM "(DTD)-bunkyo" -include 文 entity ->
<!ENTITY  % nichiji "(tuki,hi,youbi,hajime,owari)">
% person;
% bun;

<!ELEMENT kaigi(kenmei,sakusei,hakkobi,shusai,kaisaibi,basho,sanka*,naiyo,biko?)
 -作成者、発行日、件名、場所、参加者 ->

<!ELEMENT kenmei %a.ph;   -件名->
<!ELEMENT sakusei %prsinfo;   -作成者->
<!ELEMENT hakkobi EMPTY   -発行日->
<!ELEMENT shusai %prsinfo;   -主催者->
<!ELEMENT kaisaibi %nichiji;   -時->
<!ELEMENT tuki RCDATA   -月->
<!ELEMENT hi RCDATA   -日->
<!ELEMENT youbi RCDATA   -曜日->
<!ELEMENT hajime RCDATA   -開始時刻->
<!ELEMENT owari RCDATA   -終了時刻->
<!ELEMENT basho RCDATA   -場所->
<!ELEMENT sanka %prsinfo;   -参加者->
<!ELEMENT naiyo (agendum)+   -内容->
<!ELEMENT agendum %a.p;   -事項->
<!ELEMENT biko %a.p;   -備考->

<!-Because this is DTD, entities need to be called using '%' but not '&' that
   is used in instances. ->
```

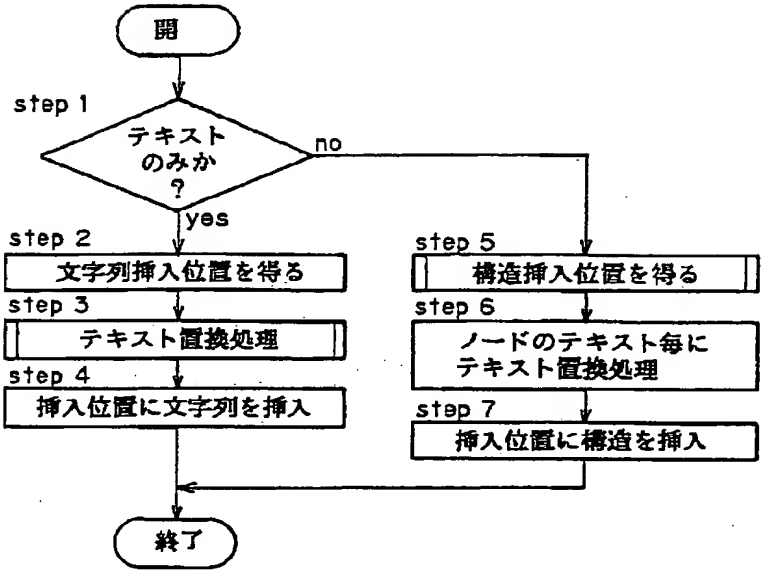
DTDの例

【図６】



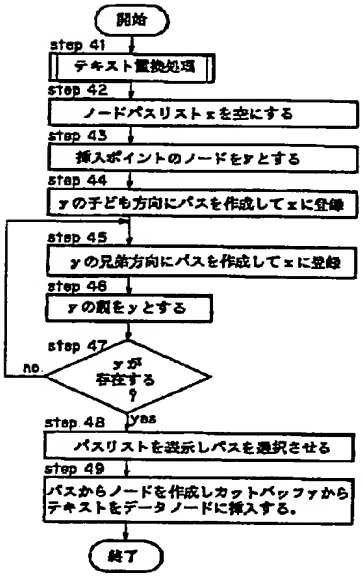
画面例

【図7】



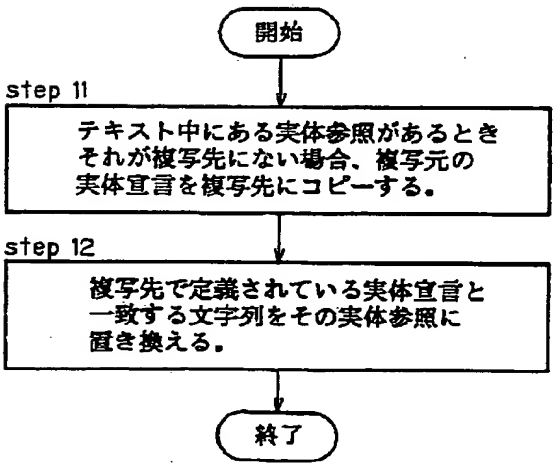
paste処理

【図10】



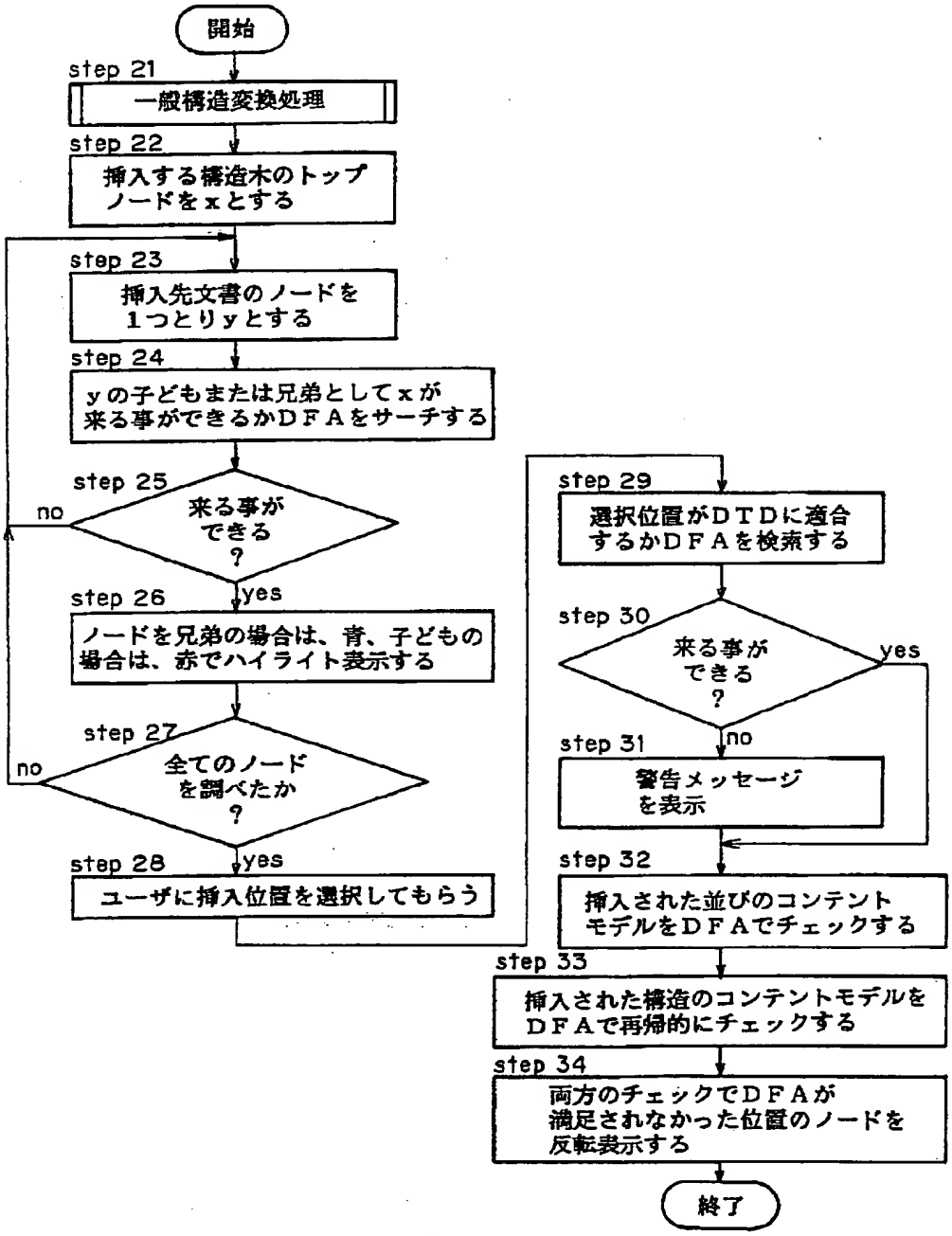
変換支援モードペースト処理

【図8】



テキスト置換処理

【図９】


開始
step 21
一般構造変換処理
step 22
挿入する構造木のトップノードをｘとする
step 23
挿入先文書のノードを１つとりｙとする
step 24
ｙの子どもまたは兄弟としてｘが来る事ができるかＤＦＡをサーチする
step 25
来る事ができる？
no
yes
step 26
ノードを兄弟の場合は、青、子どもの場合は、赤でハイライト表示する
step 27
全てのノードを調べたか？
no
yes
step 28
ユーザに挿入位置を選択してもらう
step 29
選択位置がＤＴＤに適合するかＤＦＡを検索する
step 30
来る事ができる？
yes
no
step 31
警告メッセージを表示
step 32
挿入された並びのコンテントモデルをＤＦＡでチェックする
step 33
挿入された構造のコンテントモデルをＤＦＡで再帰的にチェックする
step 34
両方のチェックでＤＦＡが満足されなかった位置のノードを反転表示する
終了

構造挿入位置